LUND UNIVERSITY

SCHOOL OF
ECONOMICS AND
MANAGEMENT

Master's Programme in Data Analytics & Business Economics

A Mixed Time-Series & Machine Learning Approach for Price Forecasting in the Swedish Ancillary Market

by

Daniel Arredondo
(da3534ar-s@student.lu.se)

DABN01
Master's Thesis (15 credits ECTS)
May 2023
Supervisor: Luca Margaritella
Word Count: 5,991

Abstract

This study aims to forecast the Swedish FCR-D Down A2 market prices through a hybrid model combining a volatility model and a machine learning approach, and compares its performance with a standalone machine learning model. We further examine the impact of different lag orders (1-Hr vs. 24-Hr) on volatility estimates and forecast performance. Evaluation metrics include root of mean squared error (RMSE) and mean absolute error (MAE). Results suggest that the hybrid model incorporating day-to-day volatility changes effectively predicts market price spikes, providing a competitive edge for energy traders. However, this model falls short during docile market periods, indicating a need for supplementary models. The study concludes that the current hybrid model does not yield significant forecast performance increases compared to simpler machine learning models, prompting further investigation into the optimization of such hybrid models for effective market price forecasting.

Keywords: Hybrid model, Volatility model, Machine learning model, Price spike prediction, Energy trading, Time series forecasting, Ancillary service market

Acknowledgements

I would like to thank my family for supporting me through out all my life. Thank you to all the students in the DABE program for being my rivals and companions in class. Thank you to the people at Modity for allowing me to be apart of their office culture.

Lastly, thank you New Zealand for being the country where I found Linnea and where my life took a turn for the better.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Research Problem

The Swedish energy system is undergoing a rapid transition towards increasing the share of renewable electricity production. While this shift brings numerous environmental benefits, it also presents significant challenges in balancing the power system due to the variable nature of renewable energy sources (Mureddu et al. (2015)). Consequently, there is a growing demand for flexible services to help mitigate these imbalances in the grid to ensure the stability and reliability energy delivery. Such services are called ancillary services whose purpose are to support power system reliability and perform the necessary services that the energy market cannot provide (Ela et al. (2018)). In Sweden, Svenska Kraftnät serves as the transmission system operator (TSO) for electricity, and they are responsible with maintaining a constant balance between energy consumption and production to keep the grid frequency at the standard level of 50 Hz.

To ensure operational reliability and manage fluctuations in power supply and demand, Svenska Kraftnät purchases different types of reserves from BRPs within the ancillary service market. One of these reserve types is the Frequency Containment Reserve (FCR), which plays a crucial role in stabilizing the frequency in the power system in response to any deviations. The procurement of FCR is divided into three ancillary service markets: Normal (FCR-N), Disturbance - Up regulation ("FCR-D Up"), and Disturbance - Down regulation ("FCR-D Down"). FCR-N is a constant, automatic service that helps stabilize the grid frequency for small changes in consumption or production, thus protecting from both positive and negative deviations from 50 Hz. Similarly, FCR-D is an automatic service that swiftly responds to large positive (negative) disturbances, where BRPs need to reduce (increase) production.

The FCR-D Down market was recently brought online at the start of 2022. Since then, the amount of procured reserves has increased quarterly in response to available and pre-qualified volume supplied by BRPs, per Svenska Kraftnät. The FCR-D Down market's design is provided in table 1.1:

TSO procures ancillary services from each market separately for each hour of the day through two auctions. The first auction (A1) closes two days prior to operation while the second auction (A2) closes the day prior. In these auctions, BRPs submit asking bids for each hour of day for energy volume (MW) and the price rate of Euros to megawatts. The TSO fulfills the cheapest bids first (i.e. merit

Table 1.1: FCR-D Down market design

System Design	**Choice**
Currency	Traded in Euros
Hourly volume requirement:	Up to approx. 538 MW
Hourly volume procurement:	Changes quarterly. As of 2023 Q2: 255 MW
Bidding Frequency:	Auction 1 occurs two days before activation Auction 2 occurs one day before activation
Minimum bid size:	0.1 MW
Pricing rule:	Pay-as-bid
Market Clearing Condition:	Merit order (ascending order of price)
Market information provided to BRPs:	Procured volume & weighted-average prices via Mimer.se
Activation:	Automatic linear activation in Hz range $50.1 - 50.5$
Activation time:	50% witin 5 seconds and 100% within 30 seconds
Endurance:	Minimum 20 minutes

ordering) under a pay-as-bid market structure to satisfy the minimum daily reserve requirements. Given the pay-as-bid structure, there does not exist a set market clearing price. Consequently, price rates for procured volumes are published as an aggregated weighted average of all satisfied bids for each hour.

As will be discussed later, the amount of MW procured and price rate for each unit is highly volatile and difficult to forecast. Additionally, the FCR-D Down market was established in the start of 2022, so there is a lack of historical data. The TSO also plans to adjust the total amount of procured reserves every quarter.

This paper focuses on forecasting the Swedish FCR-D Down A2 market prices via a combination of a volatility model and machine learning model, and compare its performance to that of a standalone machine learning model. Thus, for ease of discussion the aggregated A2 market prices will be referred to as simply the market prices. The benefits gained by an energy trader by accurately predicting the market price are competitive advantage in placing successful bids along with strategic bidding of energy volume types to ensure a price above the marginal cost of production. Additionally, this paper will make a comparison between volatility estimates calculated by two different lag orders (1-Hr lag vs. 24-Hr lag), and the resulting forecast performance improvements over each other. This is motivated by the long 24-hour forecast horizon the predictions are made on, with the assumption that day-over-day changes will carry greater volatility information than hour-over-hour changes in regards to measuring intraday price spikes. The Performance of all models will be evaluated side-by-side using the root of mean squared error (RMSFE) and mean absolute error (MAE) to confirm if such configuration of time series data and models improve forecast performance over a standalone machine learning model.

I find that the chosen artificial neural network model combined with a volatility estimate of day to day changes of market prices predicts market prices during a price spike better than the other measured models. However, the utility of the model diminishes during period of docility in market prices, and so the predictions would be best aggregated with another model capable of capturing docile market prices better.

The paper will be organized as follows. Section 2 will review related work on ancillary markets and the use of hybrid models. Section 3 will introduce a market price analysis, a description of features, and the hybrid model design. Section 4 contains the analysis of the forecasts. Section 5 concludes the paper.

2

Related Work

Performing analysis with the combination of parametric and non-parametric models is not a new concept. Donaldson and Kamstra (1997) introduces a nonlinear semi-non-parametric model built by combining the structure of the sign-autoregressive conditionally heteroskedastic (ARCH) model by Glosten et al. (1993) with that of an artificial neural network [1] to forecast stock return volatility in several markets. Zhang (2003) also uses a hybrid model assembled by combining an autoregressive-integrated-moving average (ARIMA) model and artificial neural network to capture the linear and non-linear component of a time series, respectively. They concluded that the hybrid model outperformed each component model used in isolation. On a similar note, Liu et al. (2022) used a hybrid model consisting of two non-parametric models, a weighted K-nearest neighborhood (WKNN) and Gaussian process regression, to detect electricity market price spikes and predict market prices. Two subsets of the data were created by the WKNN model identifying price spike indicator values, for which the Gaussian process model forecasted market prices conditionally on.

Several papers use machine learning models to model and forecast electricity market prices. Giovanelli et al. (2017) compares several models on the Finland's Fingrid ancillary market to forecast FCR-N market prices. They concluded that gradient boosted trees performed the best against linear regression, support vector regression, regression tree, and ARIMA(1,1,1) models for each month of 2016. Giovanelli et al. (2018) extends the prior study and compares SVR and ARIMA model to an artifical neural network, finding that the neural network performed best in most months during the year 2016. In the paper by Kraft et al. (2020), FCR market prices for the largest European markets were forecasted using the seasonal ARIMA process with exogenous variables model (SARIMAX) and artificial neural network model. They concluded that an optimized neural network outperformed the SARIMAX model, and that artificial networks benefited from an expanding training window approach. Time series analysis has also been considered when forecasting electricity market prices. Nogales et al. (2002) compares a dynamic regression model and a transfer function model to capture the serial correlation present in price forecast errors of the Spanish and California electrical market.

[1] Introduced in Section 2, Equation 7, Page 23

3

FCR-D Down Price Forecasting

3.1 A2 Market Pricing

The analysis of the FCR-D A2 market prices consists of 10,176 data points covering over one year's worth of hourly time series data (January 2022 - February 2023). The data was collected from several open data sources apart from a few weather forecast variables which were received from an energy trading company. The market prices are represented in Euros. Figure 3.1 depicts the movement of FCR-D Down A2 market prices over the time frame of interest. Large variances in market prices can be see in the months surrounding May 2022, October 2022, and January 2023. The descriptive statistics of the market prices are found in Table 3.1.

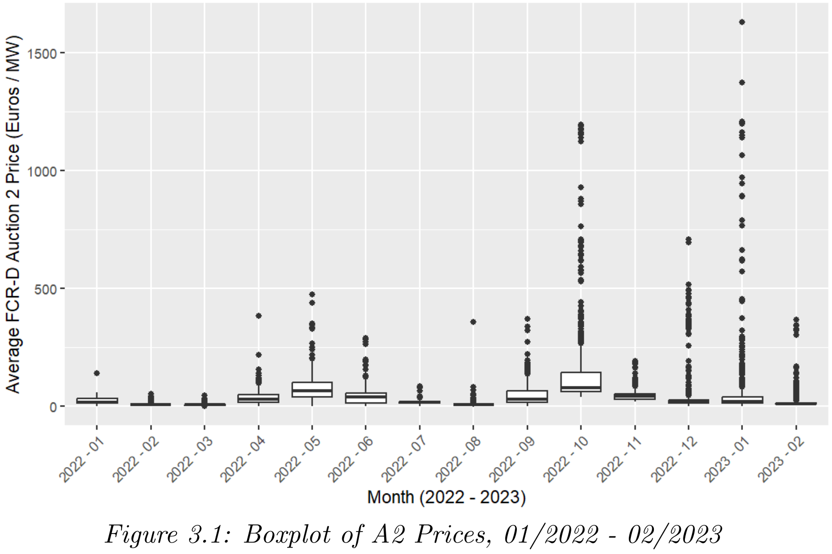


Figure 3.1: Boxplot of A2 Prices, 01/2022 - 02/2023

FCR-D is intended to automatically activate in order to control positive frequency deviations. If the TSO happens to procure all of the ancillary volume it needs for operation in the first auction, then there will be missing values in the second auction due to the lack of action by the TSO. This explains the approximately 2.4% of missing A2 market price data. The missing values have been replaced with the values of zero to ensure the continuity of the data and because the intuition remains the same. Also, volatile periods have been observed surrounding the quarterly changes in volume with persistent behavior which is seen in the large variances appear to follow quarter-endings (Apr-2022, Sep-2022, Jan-2023), possibly due to

Mean	44.797
Median	19.930
Standard Deviation	91.592
Sample Variance	8389.114
Kurtosis	86.275
Skewness	7.941
Minimum	0.000
Maximum	1630.150
% of Zeros	2.394

Table 3.1: Descriptive Statistics of FCR-D Down A2 Market Prices for Jan-2022 / Feb-2023

the changes in the amount of energy volume procurement. There exists mild negative correlations between FCR-D A2 prices and hydro energy production. According to Energiföretagen, reservoir storage levels dropped from 60% in January 2022 to below 30% in April 2022, and climbed back to about 80% in July 2022.

3.1.1 Price Analysis

I compute two time series $r_{h,t}$ by taking the first differences of A2 market prices S_t with two different lag orders h. The hourly changes, $h = 1$, and daily changes, $h = 24$, are calculated as

$$r_{h,t} = S_t - S_{t-h}. \tag{3.1}$$

The series are visualized in Figures 3.2 and 3.3. The descriptive statistics for both series are found in Table 3.2.

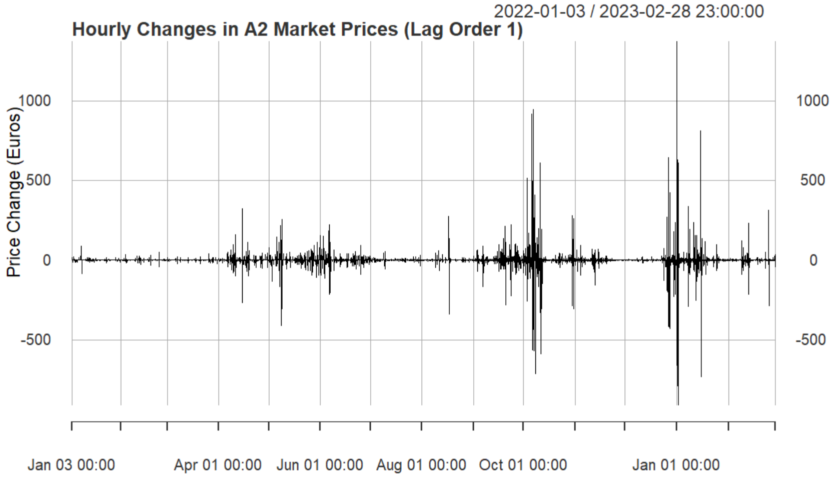


Figure 3.2: Hourly Changes, $r_{1,t}$

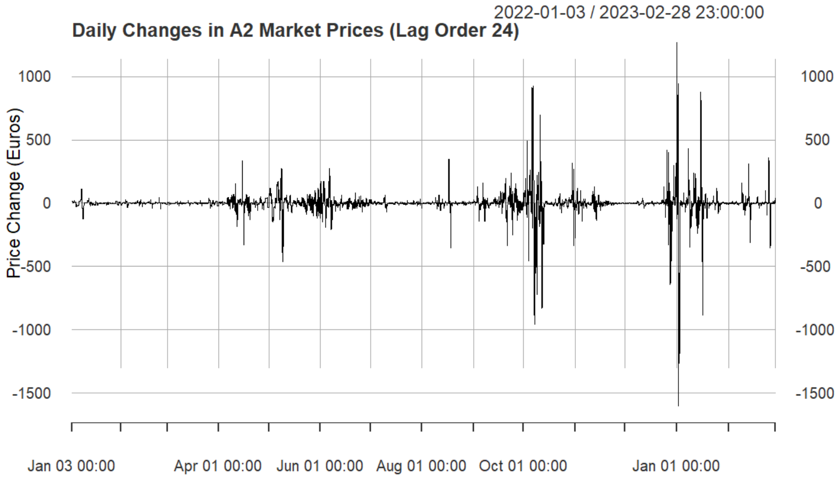


Figure 3.3: Daily Changes, $r_{24,t}$

Statistic	$r_{1,t}$	$r_{24,t}$
Mean	-0.001	-0.060
Median	0.000	-0.130
Standard Deviation	41.437	96.566
Sample Variance	1716.989	9324.988
Kurtosis	296.437	71.868
Skewness	3.666	-1.187
Minimum	-911.28	-1601.36
Maximum	1372.86	1267.73

Table 3.2: Descriptive Statistics of $r_{1,t}$ and $r_{24,t}$ for Jan-2022 / Feb-2023

Note, the log of differences was not computed like in typical time series approaches on financial market prices due to the existence of zero values in the prices. The motivation behind creating two time series for this analysis is that since predictions are made for 24 hours ahead, then there are two potential ways of viewing changes with the market prices; changes can be analyzed either in daily or hourly intervals. I perform two tests on each time series to test for stationarity and unit roots: the augmented Dickey-Fuller (ADF) test for stationarity and the Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test for level stationarity. The calculated p-values can be found in table 3.3.

Stationarity Test	$r_{1,t}$ p-values	$r_{24,t}$ p-Values
ADF Test	< 0.01	< 0.01
KPSS Test	> 0.1	> 0.1

Table 3.3: Unit Root and Stationarityy Tests. Estimates for p-values were returned by R-program functions, provided by the tseries package.

We can see that for a chosen alpha level of 0.5, the two time series exhibit stationary characteristics. The ADF test results in p-values less than 0.01 for both series, resulting in the rejection of the null hypothesis of a unit root present in the series. The KPSS test results in p-values greater than 0.1, resulting in the failure to reject the null hypothesis of the series being stationary.

3.2 Feature Selection

When modeling machine learning algorithms, it is important to use features that are related to the dependent variable at hand. Thus, the following types of data have been chosen to use in modeling the A2 market price:

- FCR-D market data, such as A1 price and volumes and spot pricing
- Renewable energy production, such as hydro production
- Day relative items, such as holiday, weekend, hour of day, etc.

The feature types are broken down in more detail in Table 3.4:

Table 3.4: Selected Features for Price Forecasting

Feature Name	Number of Features
Calendar (year, hour, holidays, etc.)	6
eSett Spot Price	1
Highly Correlated FCR-D Market Values	8
Renewable Energy Production Values	8
Volatility Forecast	1
Total Features	24

The collected features were standardized prior to modeling with the artificial neural network model, but were not pre-processed with GBRT and random forest models.

Remark. Potential Feature Bias In a time series analysis with several features/regressors, it is important to have all data appropriately aligned so as to not induce any bias of future values on forecast predictions. In this paper an assumption is made that since market prices are forecasted a day prior to fulfillment, then energy traders will have access to highly accurate renewable energy production forecasts. Thus, when training the models in this paper, actual energy productions values were substituted for forecast values to make market price predictions of the same time frame.

3.3 Model Design

3.3.1 Hybrid Modeling

An axiomatic saying is that there is no perfect model to answer every problem. Page (2018) states the motivation behind *many-model thinking* is that it produces wisdom through a "diverse ensemble of logical frames." Herein lies the main motivation behind the usage of the following hybrid model structure.

The intent of the hybrid model used in this paper is to use a traditional, parametric model to fit and forecast volatility in market prices, and implement the results in a non-parametric model to improve forecast performance. The volatility forecast is intended to be a strong explanatory variable for modeling market prices among other variables when inputted through a regression-based machine learning model. The hybrid structure used in this paper is shown in Figure 3.4.

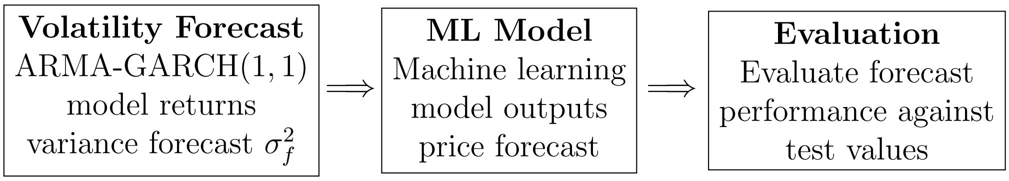


Figure 3.4: Proposed Hybrid Model Structure

The model design and methodology used for forecasting FCR-D Down market prices will be explained in depth in the following sections.

3.3.2 Volatility Modeling

I test for heteroskedastic volatility in $r_{1,t}$ and $r_{24,t}$ by following the process laid out in Enders (2009)[1] by first fitting an Autoregressive-Integrated-Moving Average (ARIMA) process to both time series and then analyzing the ACF correlogram of the squared residuals of both processes. The ARIMA process was fitted on $r_{1,t}$ and $r_{24,t}$ using the *auto.arima* function in the R programming package *forecast*. The resulting ACF plots are shown in Figures 3.5 and 3.6, with dotted red lines at every 12 lags.

We can see in Figure 3.5 there are significant autocorrelations at the first and 14th lag. In Figure 3.6, there are significant autocorrelations at the first and 24th lag. This is indicative of an underlying autoregressive conditionally heteroskedastic (ARCH) process in our two time series. Since I detected an ARCH process in

[1] Page 129

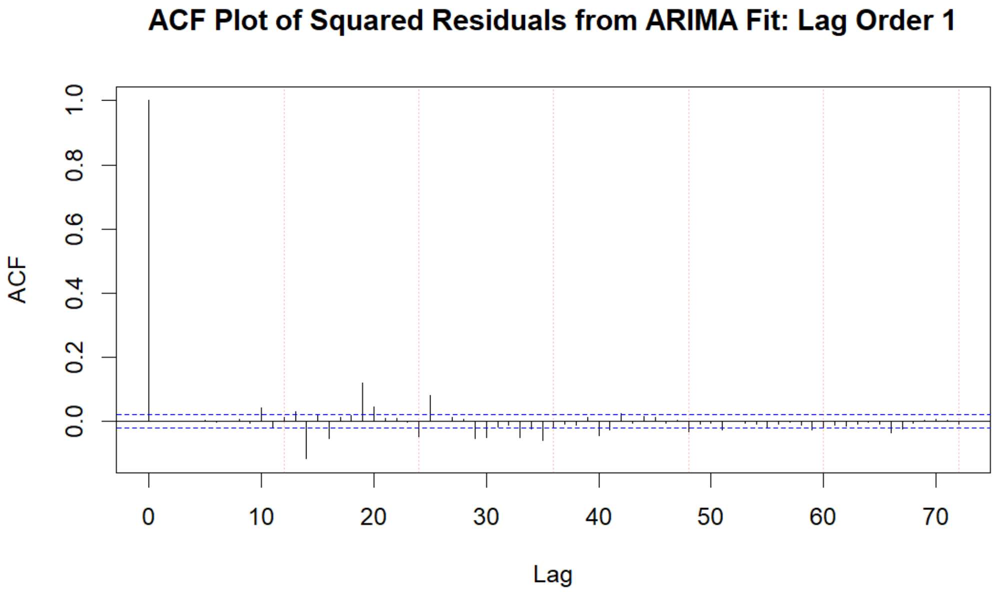


Figure 3.5: ACF correlogram for the squared residuals from the fitted ARIMA on $r_{1,t}$

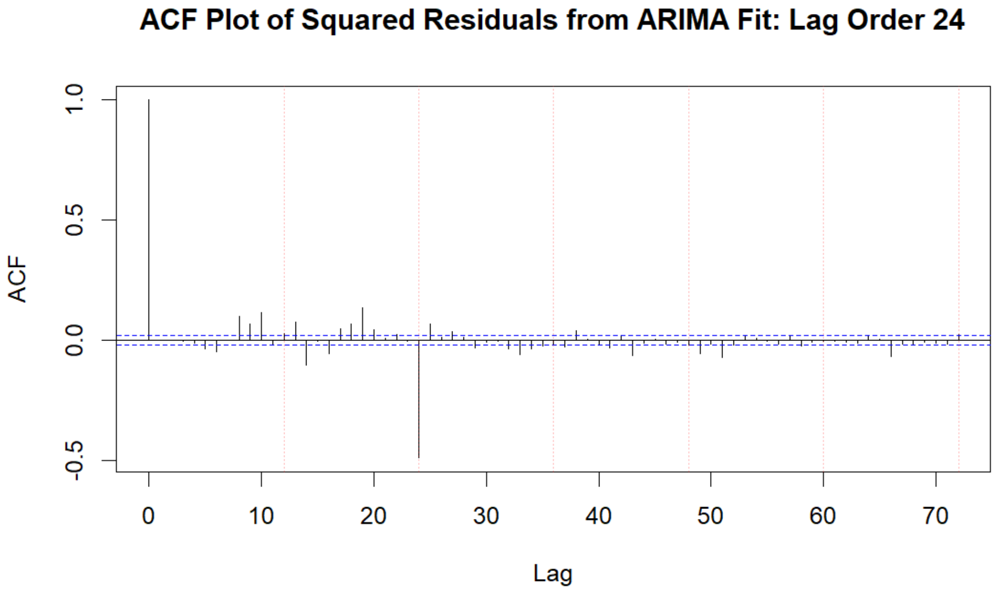


Figure 3.6: ACF correlogram for the squared residuals from the fitted ARIMA on $r_{24,t}$

both $r_{1,t}$ and $r_{24,t}$, I assume that the conditional mean forecast of $r_{1,t}$ and $r_{24,t}$ is $\mathbb{E}(r_{1,t}|I_{1,t-1})$ and $\mathbb{E}(r_{24,t}|I_{t-1})$, respectively, and is expressed as the following:

$$r_{h,t} = \mathbb{E}(r_{h,t}|I_{h,t-1}) + \varepsilon_{h,t}, \qquad h = 1, 24 \tag{3.2}$$

where $\varepsilon_{h,t}$ represents the residuals and $I_{h,t-1}$ represents the information set of prior market price changes. I also assume the forecast residuals $\varepsilon_{h,t}$ have zero mean and conditional variances represented as

$$\mathbb{E}(\varepsilon^2_{h,t}|I_{h,t-1}) = \sigma^2_{h,t} \qquad h = 1, 24 \tag{3.3}$$

The aim of selecting a conditional volatility model is to produce an accurate forecast of $\sigma^2_{1,t}$ and $\sigma^2_{24,t}$ to include in the data prior to the processing by the non-parametric model. Thus, I chose to forecast $\sigma^2_{1,t}$ and $\sigma^2_{24,t}$ with a Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model by Bollerslev (1986), which has the following framework:

$$\sigma^2_t = \alpha + \sum_{i=1}^{p} \beta_i \sigma^2_{t-i} + \sum_{j=1}^{q} \gamma_j \varepsilon^2_{t-j}, \tag{3.4}$$

where α represents the sample mean of σ^t_t, and p, q represent the number of lags to include of σ^2_t and ε^2_t, respectively. I chose this model due to its wide-usage in econometrics analysis. The model also is able to fit and forecast time series data whose variance appears to be time dependent, or in other words the model assumes that the conditional volatility of the time series is dependent on previous observations, which in turn allows it to capture periods of volatility and docility. This is equivalent to the assumptions made in Equation (3.3). In this paper, I model $r_{h,t}$ with an ARMA-GARCH(1,1) process. The ARMA-GARCH process is defined as follows for $r_{h,t}$:

$$\begin{aligned} r_{h,t} &= E(r_{h,t}|I_{h,t-1}) + \varepsilon_{h,t} \qquad h = 1, 24 \\ &= \mu_h + a_h r_{h,t-1} + b_h \varepsilon_{h,t-1} + \varepsilon_{h,t}, \end{aligned} \tag{3.5}$$

$$\varepsilon_{h,t} \sim \sigma_{h,t}\epsilon_{h,t}, \qquad \epsilon_{h,t} \sim^{i.i.d.} \mathcal{N}(\mu_\epsilon = 0, \sigma_\epsilon = 1) \tag{3.6}$$

$$\begin{aligned} \sigma^2_{h,t} &= E[(r_{h,t} - \mu_{h,t})^2] \\ &= \beta_h \sigma^2_{h,t-1} + \gamma_h \varepsilon^2_{h,t-1} \end{aligned} \tag{3.7}$$

where μ_h is the mean of $r_{h,t}$, $\{a_h, b_h\}$ represent the coefficients for the ARMA process on the conditional mean of $r_{h,t}$, and $\{\beta_h, \gamma_h\}$ represent the coefficients for the GARCH process on the conditional variance of $r_{h,t}$.

With the above framework, our volatility forecasts take the following form:

$$\hat{\sigma}^2_{h,t+1} = \hat{\beta}^{(h)} \sigma^2_{h,t} + \hat{\gamma}^{(h)} \varepsilon^2_{h,t} \tag{3.8}$$

where $\hat{\beta}^{(h)}, \hat{\gamma}^{(h)}$ are maximum likelihood estimates respective to the lag order h calculated via the R programming package *rugarch* by Ghalanos (2022).

3.3.3 Regression Modeling

An ideal scenario in the FCR-D market would be that all information of market players and their respective bids is known, so as to be able to ”reverse-engineer” an algorithm that captures the behavior of market pricing. In such ideal scenario, the reserve supply function equilibrium would be calculable by an energy trader, and lead to strategically placed bids. Since such a scenario does not exist, a regression analysis is performed on the collected data to find a relationship between A2 market prices and regressors. A regression analysis is a statistical tool used to investigate relationships between a dependent variable y_t and independent variables $\{x_{1,t}, \ ..., \ x_{n,t}\}$ for $t = 1, ..., T$. In the context of this paper, t represents time in hours, $x_{i,t}$ represents the i-th independent variable at time t for $i = 1, \ ..., \ 24$. A simple linear regression with n independent variables can be modeled as

$$y_t = \beta_0 + \beta_1 x_{1,t} + ... + \beta_n x_{n,t} + \varepsilon_t, \qquad t = 1, \ ..., \ T$$

which then by ordinary least squares (OLS) we can estimate the coefficients $\{\beta_0, \beta_1, ..., \beta_n\}$. The estimated coefficient values $\{\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_n\}$ can be considered as measures of explanatory power that each independent variable $\{x_{t,1}, x_{t,2}, ..., x_{t,n}\}$ has on y_t.

The estimated coefficients can also then be used to forecast the value of y_{t+1} with the introduction of new data $\{x_{1,t+1}, x_{2,t+1}, ..., x_{n,t+1}\}$ such that

$$y_{t+1} - \varepsilon_{t+1} = \hat{y}_{t+1} = \hat{\beta}_0 + \hat{\beta}_1 x_{1,t+1} + ... + \hat{\beta}_n x_{n,t+1}.$$

A measure of forecast performance can then be calculated, which will be measured in this paper via the root mean square error (RMSE) metric. I use the RMSE metric to leverage its sensitivity to outliers in the data, which in our case would be price spikes. Given a 24 element vector of forecasts $\hat{\boldsymbol{y}} = [\hat{y}_{t+1}\ \hat{y}_{t+2}\ ...\ \hat{y}_{t+24}]^\top$, this metric is written as,

$$\sqrt{MSE(\hat{\boldsymbol{y}})} = \sqrt{\frac{1}{24}\sum_{i=1}^{24}(y_{t+i} - \hat{y}_{t+i})^2}. \tag{3.9}$$

In this paper I explore a few machine learning regression models to learn a function $F : \mathbb{R}^{24} \to \mathbb{R}$ to obtain predictions such that,

$$F(\boldsymbol{x}_{t+1}) = \hat{y}_{t+1} = y_{t+1} - \varepsilon_t + 1,$$

where $\boldsymbol{x}_{t+1}$ represents a vector of new data input, y_{t+1} the A2 market price at time $t+1$, and $F(\boldsymbol{x}_{t+1}) = \hat{y}_{t+1}$ the predicted market price at time $t+1$.

Decision Trees and Ensemble Methods

A decision tree is a capable rule-based, supervised machine learning algorithm that maps an input space to ”leaves”, or disjoint regions defined by rules or restrictions on regressors. Each leaf then has a constant value as an output, which in our case of market prices, an averaged market price. These decision trees with continuous output spaces are considered regression trees. The advantage of using a decision tree is that it is a highly interpretable algorithm, ideal for a business environment. A predictive decision tree can be generalized mathematically as,

$$\hat{y}_t = \sum_{l=1}^{L} \hat{y}_l\ I\{\boldsymbol{x}_t \in R_l\}, \tag{3.10}$$

where L is the total number of leaf regions in the decision tree, R_l is the lth region, I is the indicator function, and $\hat{y}_l$ is the constant, real-valued prediction assigned to the lth region. This paper introduces and utilizes two sophisticated regression tree-based algorithms: Gradient Boosted Regression Trees (GBRT) and Random Forests, both of which are forms of ensemble learning methodologies. GBRTs are a distinct category of ensemble learning algorithms, characterized by the sequential training of simple decision trees, known as ’weak learners’. Each tree is meticulously trained to focus on the residuals, $y_{t+1} - \hat{y}_{t+1}$, left by its predecessor. This systematic approach of adding new models to correct the errors made by existing models facilitates a reduction in overall model bias while simultaneously maintaining a low model variance. Among the various forms of GBRTs, this study employs an advanced version, known as the Extreme Gradient Boosted Regression Tree (XGBoost) by Chen and Guestrin (2016). XGBoost is extensively recognized and utilized in numerous machine learning competitions and analytical procedures due to its inherent ability to include regularization in the learning process, thereby effectively controlling overfitting to the training data. Random Forests, another type of ensemble learning algorithm, construct and aggregate the predictions of multiple regression trees. Contrary to the sequential training process of GBRTs, decision trees in a Random Forest are trained in parallel. Each tree is independently built using a random subset of the predictors and a bootstrapped sample of the training data. The randomness induced in the feature selection and data sampling processes effectively prevents overfitting to the training data, thereby improving the model's generalization capabilities. The *XGBoost* Python package will be employed through the Scikit-Learn API to train both the GBRT and Random Forest models in this study. One significant advantage of using this package is its capacity to provide highly optimized implementations of gradient boosting and decision tree learning algorithms. This enables a significant reduction in processing time, thereby increasing the computational efficiency of the model-building process. According to the *XGBoost* documentation, *XGBoost* uses 2nd order approximation to the objective loss function. In comparison, other typical packages used to learn Random Forest models use exact approximation, leading to increased computation times. Unfortunately, another caveat with the package is that bootstrap sampling is not available to use with random forest model training, so different ratios of data subsampling is instead used to tune the model to control overfitting.

Artificial Neural Networks

ANNs are composed of interconnecting artificial neurons (also called nodes or units) which are organized in layers. These networks can model intricate, non-linear relationships that may exist in large, multi-dimensional data. The fundamental building block of an ANN, an artificial neuron or node, applies a non-linear function, typically called the activation function, to the weighted sum of its inputs. The network is made up of multiple layers of these neurons—input layers, hidden layers, and an output layer. The hierarchical structure allows an ANN to transform the input in a way that makes it possible to learn and represent complex patterns.

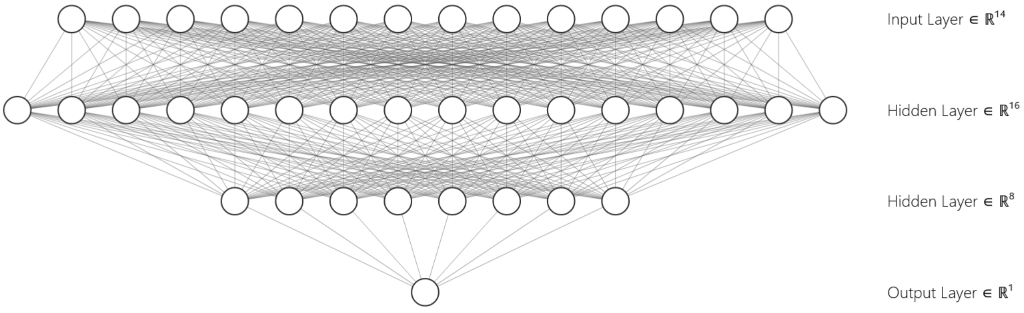

Input Layer $\in \mathbb{R}^{14}$
Hidden Layer $\in \mathbb{R}^{16}$
Hidden Layer $\in \mathbb{R}^{8}$
Output Layer $\in \mathbb{R}^{1}$

Figure 3.7: Example of ANN architecture

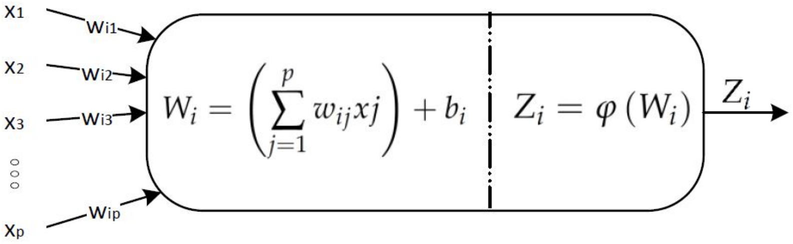

X1
Wi1
X2
Wi2
X3
Wi3
Xp
Wip
$W_i = \left(\sum_{j=1}^{p} w_{ij}xj\right) + b_i$
$Z_i = \varphi\left(W_i\right)$
Z_i

Figure 3.8: Structure of ANN neuron by Giovanelli et al. (2018), where $\{x_j\}_{j=1}^{p}$ represents p input neurons from the prior layer, $\{w_{ij}\}_{j=1}^{p}$ represents the weights for the i-th neuron, b_i represents the bias term, and ρ represents the non-linear function

One of the unique characteristics of an ANN is its ability to learn from the data. During the training process, the network adjusts the weights of the connections based on the error it made in its predictions, striving to minimize this error. This learning procedure, often called backpropagation, allows the network to map inputs to the right outputs.

The activation function chosen in this paper is a rectified linear unit (ReLU) function due to its popularity, which is defined as ReLu : $\rho(x) = \max(0, x)$, for $x \in \mathbb{R}$. The The design of the ReLU function helps to mitigate the vanishing gradient problem, which is a difficulty encountered when training neural networks during the backpropagation process. This problem arises when the gradients, or partial derivatives, of the loss function with respect to the model parameters become minuscule. Parameter updates are proportional to the size of the gradients, meaning that learning can become slow or stop altogether. Because the derivative of the ReLU function is always 1 for positive inputs, this helps ensure that the gradients do not vanish during training.

The ANN model used in this paper also includes a dropout layers. Dropout is a regularization technique used in ANNs to prevent overfitting. It was introduced by Srivastava et al. (2014) and has since become a standard method in training deep learning models.

During training, dropout randomly selects a subset of neurons and temporarily ”drops out” or deactivates them, along with all their incoming and outgoing connections. The proportion of neurons dropped out is a hyper-parameter and is usually set to a value between 0.2 and 0.5; that is, during each training stage, each neuron is set to zero with that probability.

The stochastic process within dropout layers makes the model more robust because it forces the model to learn multiple independent representations of the same data[2]. A neuron cannot rely on the presence of particular other neurons and is forced to learn more robust features that are useful in conjunction with many different random subsets of the other neurons.

At forecast time, no units are dropped out; rather, their outputs are scaled down by a factor equal to the dropout rate to take into consideration that more units are active than during training. In other words, the weights of each node are multiplied by the dropout rate.

[2] ”Dropout is a bagging-like technique that allows us to combine many neural networks without the need to train them separately” - Lindholm et al. (2022)

3.4 Empirical Model Training & Tuning

The volatility model was initialized with data covering the entirety of the months January 2022 and March 2022. As a result, the volatility forecasts extend from April 2022 to February 2023, which represents the time frame of data the regression models are trained on. Forecasts were performed using a training and evaluation method called rolling-origin-recalibration (ROR), as defined by Bergmeir and Benítez (2012). In this method, forecasts for 24-hr horizon are performed by sequentially moving values from the test set to the training set, thus changing the value of the last known time point t representing the last known set of values $\{x_{1,t},\ x_{2,t},\ ...,\ x_{n,t}\}$. For each forecast, the model is re-trained using all available data in the training set, which often means a complete retraining of the model. In other words, the training window expanded with each iteration with 24 hours worth of data, and the test window was moved ahead 24 hours. In the testing of the volatility model this implies that for each new day that passed, the ARMA-GARCH model was retrained and new values of coefficients were obtained. Consequently, this involves a higher computation cost to thoroughly measure each combination.

Each regression model was calibrated via a grid search of hyper-parameters, which is a looping through combinations of hyper-parameters within the models to identify the combination resulting in the lowest calculated RMSE, without the implementation of the volatility forecast. This calculation of RMSE was done via the previously mentioned ROR methodology. The initial window size chosen for the regression models was 180×24 hours, or about half of a year's worth of hours.

The decision to retrain the models after each day was due to the lack of historical data in the FCR-D Down market and the frequent changes in market design. In future applications, a rolling-window approach may be better suited to decrease the noise from older data stemming from different market designs. Such a method was chosen by Giovanelli et al. (2018) to analyze the Finnish Fingrid ancillary market.

3.4.1 Gradient Boosted Regression Tree Tuning

The hyperparameter tuning process on the GBRT model began the default parameters as proposed by the XGBoost package and later tuned via experimentation of combination of paramaters. The default and tuned parameters can be found in Table 3.5.

First, max tree depths and data subsampling rates were first tuned simultaneously. Then, the number of sequential trees trained (boosting rounds) was trained tuned simultaneously with the learning rate. Finally, the regularization parameters Alpha and Lambda, reflecting L_1 lasso and L_2 ridge regression regularization techniques respectively, were trained. The experimental results are shown in Figure 3.9.

3.4.2 Random Forest Tuning

Many parameters in the Random Forest model were tuned through a similar process performed with the GBRT model. The default parameters for the Random Forest model provided through the Scikit learn API and the resulting tuned parameters are listed in Table 3.6. One of the most important hyperparameters when tuning a Random Forest model is to choose how many trees will be trained, affecting the

Parameter	Default Value	Tuned Value	Description
Max Tree Depth	6	5	Maximum depth of a regression tree. Higher values increase model complexity
Subsample ratio	1	0.5	%Data randomly uniformly sampled for training. Occurs in each boosting iteration
Learning Rate	0.3	0.1	Step size shrinkage in weight updates
Number of Estimators	No Limit	20	Number of boosting rounds
Alpha Value	0	4	L_1 lasso regularization term
Lambda Value	1	4	L_2 ridge regression regularization term

Table 3.5: RMSE performance of GBRT model with varying combinations of data sub-sampling rates and max tree depths. The pair of values of max tree depth and subsampling rate the lead to the lowest RMSE value is highlighted in the plot.

total number of predictions to be aggregated. Thus, this was the first parameter tuned. In Figure 3.10a, it is seen that the RMSE reaches a minimum when the number of trees trained is equivalent to 64. After the parameter has been adjusted, I performed a grid search between the ratio of subsampled features and the ratio of subsampled data. In theory, Random Forest models incorporate bootsampling of the data, but since bootsampling is not an available option in the XGboost pacakge in Python, the ratio of subsampled data is tuned to mimic the induced randomness bootstrap sampling creates. The result of this tuning process included two pairs of ratios, feature subsampling and data subsampling, that performed similarly, which can be seen in Figure 3.10b. Lastly, the regularization values Alpha and Lambda were tuned to achieve the best measure of RMSE. By increasing these values, the model is also increasingly regularized and thus trains decisions trees that are not as deeply grown. According to Figure 3.10c, we see that changes in Lambda, the value for type L_2 regularization, affects the model more than changes in the Alpha value, the value for type L_1 regularization.

3.4.3 Artificial Neural Network Tuning

The structure of the ANN used in this paper is shown in Table 3.7. I chose an ANN with 7 hidden layers in between the input and output layer, which includes two dropout layers preceding each linear input layer. To tune the ANN model, I first experimented with grid search of hyper-parameter values between epochs and dropout rates. The reasoning is that a lower dropout rate will require a higher number of epochs to balance the increased regularization in the model. I then followed this with an experiment on the learning rate. Have a smaller learning rate will imply slower training process, which would need to be optimized since we have updated the number of epochs we train the model. The final model will have

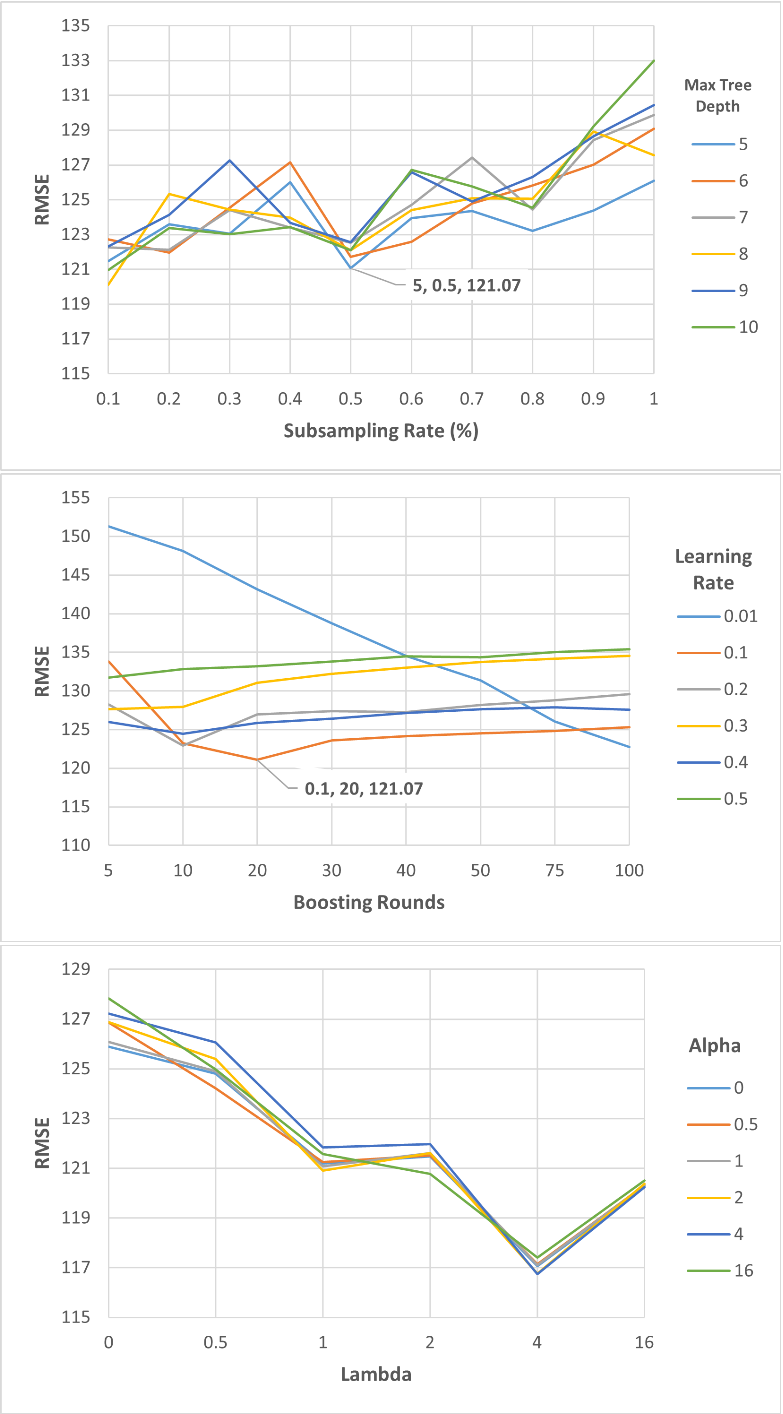
Parameter	Default Value	Tuned Value	Description
Max Tree Depth	No Limit	No Limit	Maximum depth of a regression tree. Higher values increase model complexity
Data Sampling ratio	0.8	0.7	%Data randomly uniformly sampled for training. Occurs in each tree
Feature Sampling Ratio	0.8	0.25	Features randomly uniformly sampled for training. Occurs at each node of each trained tree
Learning Rate	1	1	Step size shrinkage in weight updates
Trees in Forest	100	64	Number of Trees in the trained in parallel in Random Forest model
Alpha Value	0	16	L_1 lasso regularization term
Lambda Value	0.00001	1	L_2 ridge regression regularization term

Table 3.6: List of default and tuned parameters for Random Forest model

a dropout rate of 30%, be trained using 500 epochs, and have a learning rate of 0.0001.

Layer	Description
Input Layer	Independent variables $\{x_{i,t}\}_{i=1}^{24}$ for $t \in \{1, 2, ..., T\}$
Hidden Layer 1	Dropout layer with 30% drop rate
Hidden Layer 2	Linear regression layer
Hidden Layer 3	ReLu activation layer - 32 nodes
Hidden Layer 4	Dropout layer with 30% drop rate
Hidden Layer 5	Linear regression layer
Hidden Layer 6	ReLu activation layer - 16 nodes
Output Layer	Linear regression layer

Table 3.7: Structure of ANN used in the paper

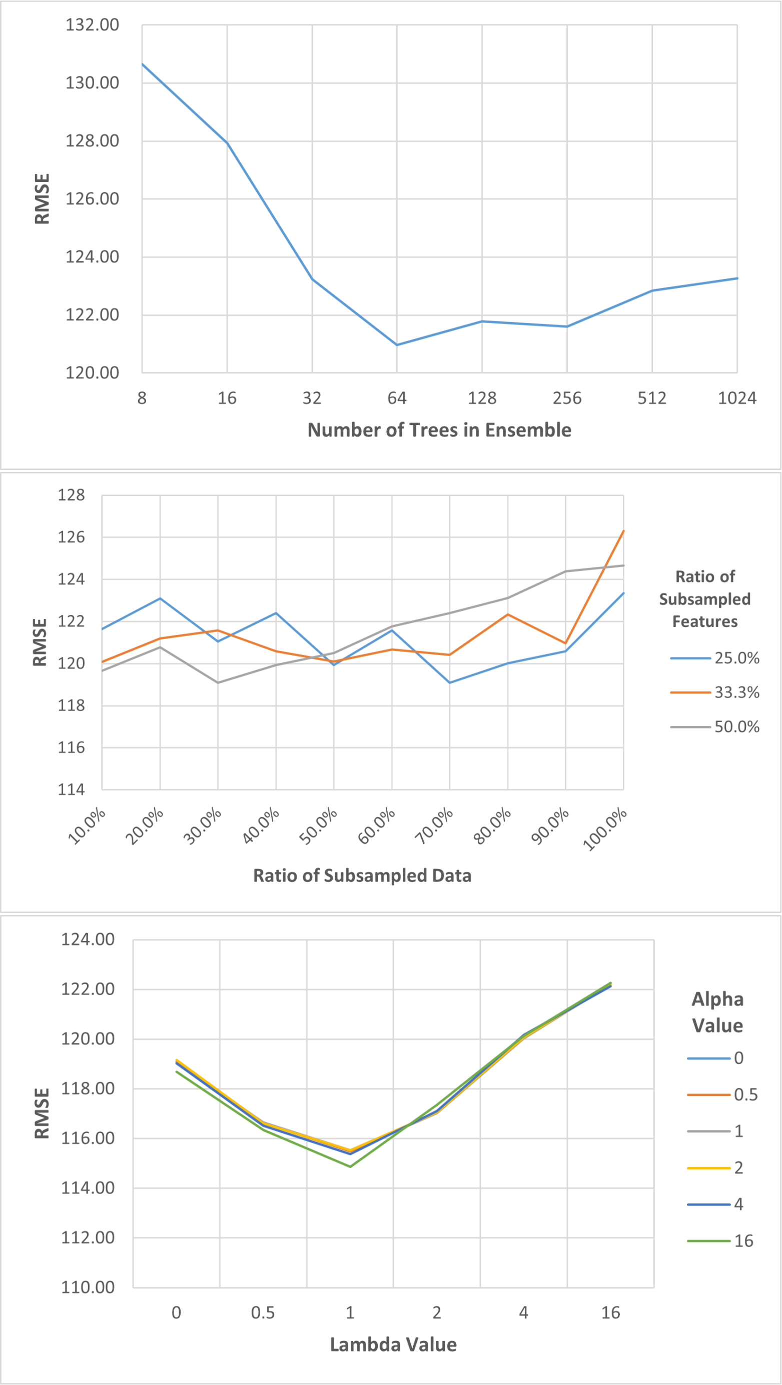


(a) RMSE performance of varying combinations of data sub-sampling rates and max tree depths. The pair of values of max tree depth and subsampling rate the lead to the lowest RMSE value is highlighted in the plot.

(b) RMSE performance of varying combinations of number trees trained sequentially trees and learning rates. It is a tradeoff between boosting rounds and learning rate, with lower rates leading to more boosting rounds to converge on the lowest RMSE resulting in higher computational costs. Thus, a value of 20 boosting rounds was chosen to keeps computation costs low, which is highlighted in the plot.

(c) RMSE performance of varying combinations of regularization parameters Alpha and Lambda. Changing Lambda resulted in a greated decrease in RMSE than changing Alpha.

Figure 3.9: RMSE performance of GBRT model with varying combinations of hyperparameters

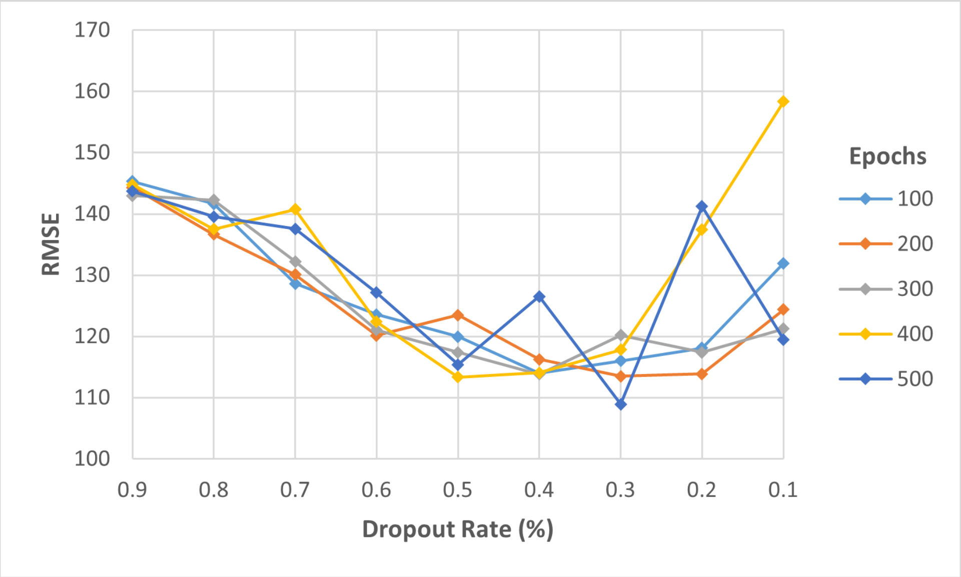


(a) RMSE performance of Random Forest model with different forest sizes of $[8, 16, 32, 64, 128, 256, 512, 1024]$ *trees. RMSE decreases before 64 trees and increases after 64 trees, meaning there is an optimal amount of aggregated predictions around 64 trees.*

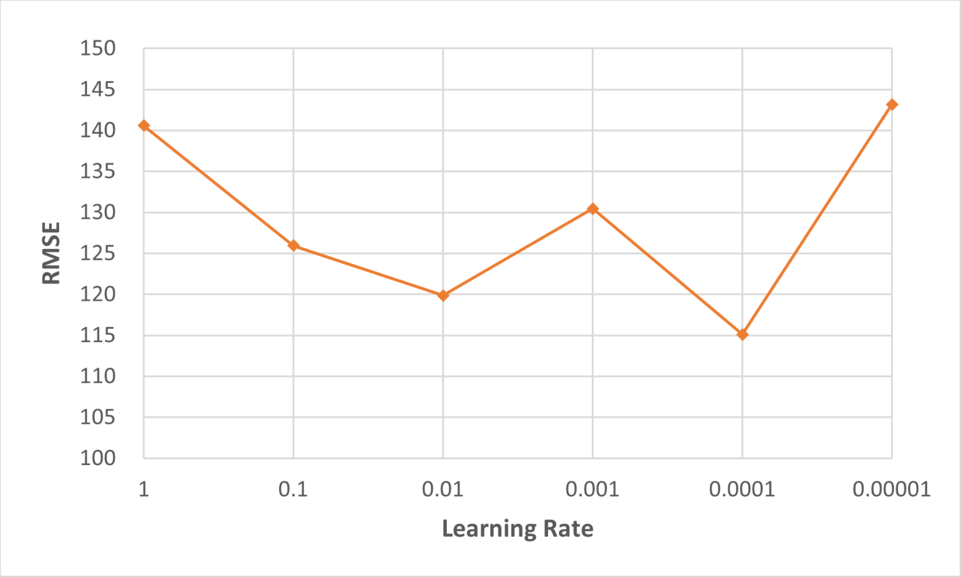
(b) RMSE performance of Random Forest model with different ratios of sub-sampled features and input data. The ratios of subsampled data followed incremental increases of 10% starting at 10% to 100%, while the ratios of subsampled features tested were $[1/4, 1/3, 1/2]$*. The chosen pair of values here was 70% of subsampled data and 25% of subsampled features.*

(c) RMSE performance of Random Forest model with different values of regularization parameters Alpha and Lambda. There was little to no improvement when adjusting the value of Alpha, but in the pursuit of a simpler model, a combination of values of 1 and 16 were chosen from Lambda and Alpha, respectively.

Figure 3.10: RMSE performance of Random Forest model with varying combinations of hyperparameters



(a) RMSE performance of varying combinations of dropout rates and epochs. RMSE value decreased as dropout rate decreased until around a value of 0.3, meaning less nodes were activated in each layer. I chose a combination of 30% dropout rate with a 500 epochs for the ANN model.



(b) RMSE performance of varying values of learning rates. After having chosen 500 epochs and a dropout rate of 30%, the best performing learning rate was a value of 0.0001.

Figure 3.11: RMSE performance of ANN model with varying combinations of hyperparameters

4

Empirical Analysis

4.1 Volatility Estimates

I have calculated the forecast estimates for the time frame of 2022-03-01 to 2023-02-28, using the initial three months of the year 2022 to initialize the window for the ARMA-GARCH model. After each forecast, the training and test set was expanded forward by 24 hours and the model subsequently retrained. The resulting volatility forecasts $\hat{\sigma}^2_{1,t}$ and $\hat{\sigma}^2_{24,t}$ are plotted against each other in Figure 4.1.

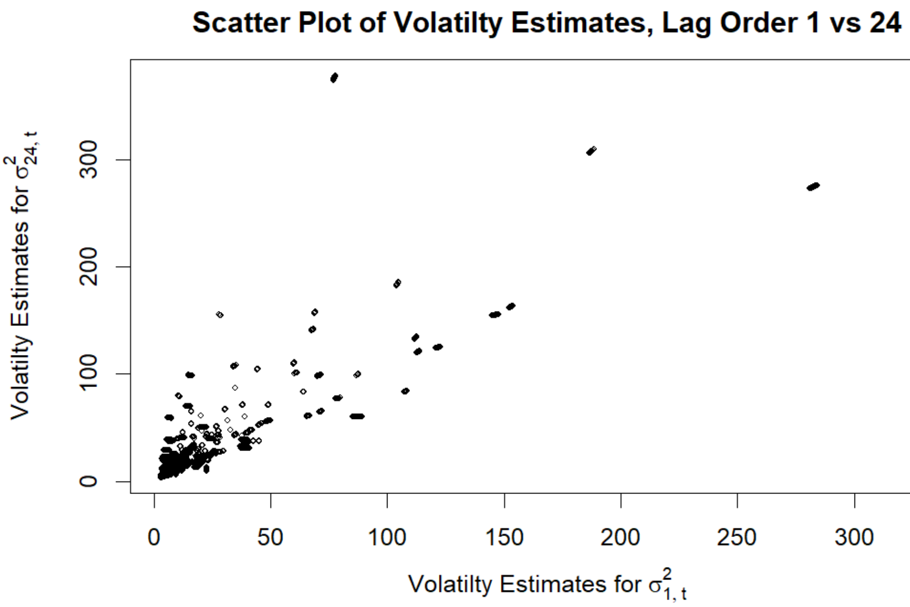


Figure 4.1: Comparison of Volatility Estimates

We see that volatility estimates for $\hat{\sigma}^2_{24,t}$ are mostly greater than the estimates calculated for $\hat{\sigma}^2_{1,t}$. According to Figure 4.2, I find that the volatility estimates $\hat{\sigma}^2_{24,t}$ are higher because they have a higher level of perseverance after a period of high volatility. Also, it appears that $\hat{\sigma}^2_{24,t}$ has higher monthly maximum and minimum values in comparison to $\hat{\sigma}^2_{1,t}$ in the data. The values are shown in Table 4.1.

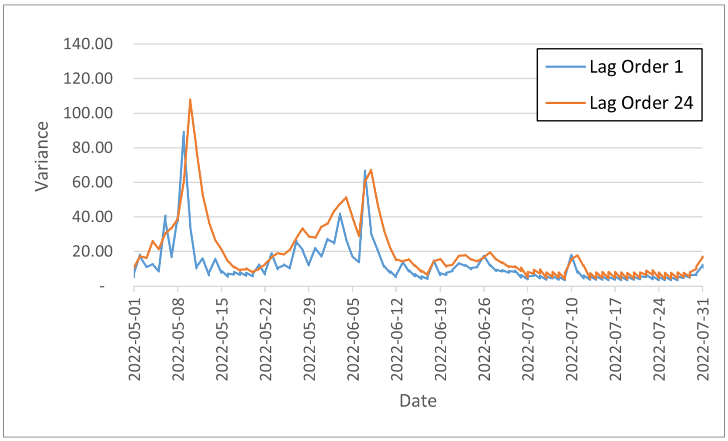


Figure 4.2: Comparison of Volatility Estimates Between 01/05/2022 and 31/07/2022

Volatility	$\hat{\sigma}^2_{1,t}$			$\hat{\sigma}^2_{24,t}$		
Month	Min	Max	Mean	Min	Max	Mean
May-22	5.18	89.39	17.27	7.91	107.98	28.48
Jun-22	4.15	66.85	15.77	6.47	67.27	24.84
Jul-22	3.31	18.1	5.89	4.2	17.87	8.23
Aug-22	3.11	49.86	7.44	3.9	65.38	12.54
Sep-22	3.85	46.18	17.36	4.27	61.45	24.85
Oct-22	4.25	284.05	44.09	5.34	309.27	57.89
Nov-22	3.76	44.53	11.98	4.65	104.65	17.91
Dec-22	3.56	153.78	18.88	4.33	163.53	23.88
Jan-23	5.00	334.24	41.21	5.71	378.26	68.27
Feb-23	4.02	64.24	14.32	4.91	87.21	21.37

Table 4.1: Range and Mean of Variance Estimates

4.2 Performance Evaluation

An initialization window of 24 × 30 hours was chosen for all regression models representing a window of 30 days. Thus, predictions for market prices begin 2022-05-01 and end at 2023-02-28. I also calculated the results of the models for three sets of data consisting of

1. data with no volatility forecast (No $\hat{\sigma}^2_{h,t}$),
2. data with $\hat{\sigma}^2_{1,t}$,
3. data with $\hat{\sigma}^2_{24,t}$.

Variable Importance

To analyze the impact of the calculated variables have made on the models, I extracted the feature importance values from both the GBRT and Random Forest models in each iteration of the expanding window, and calculated the averages for each feature. The results are shown in Figure 4.3 and Figure 4.4.

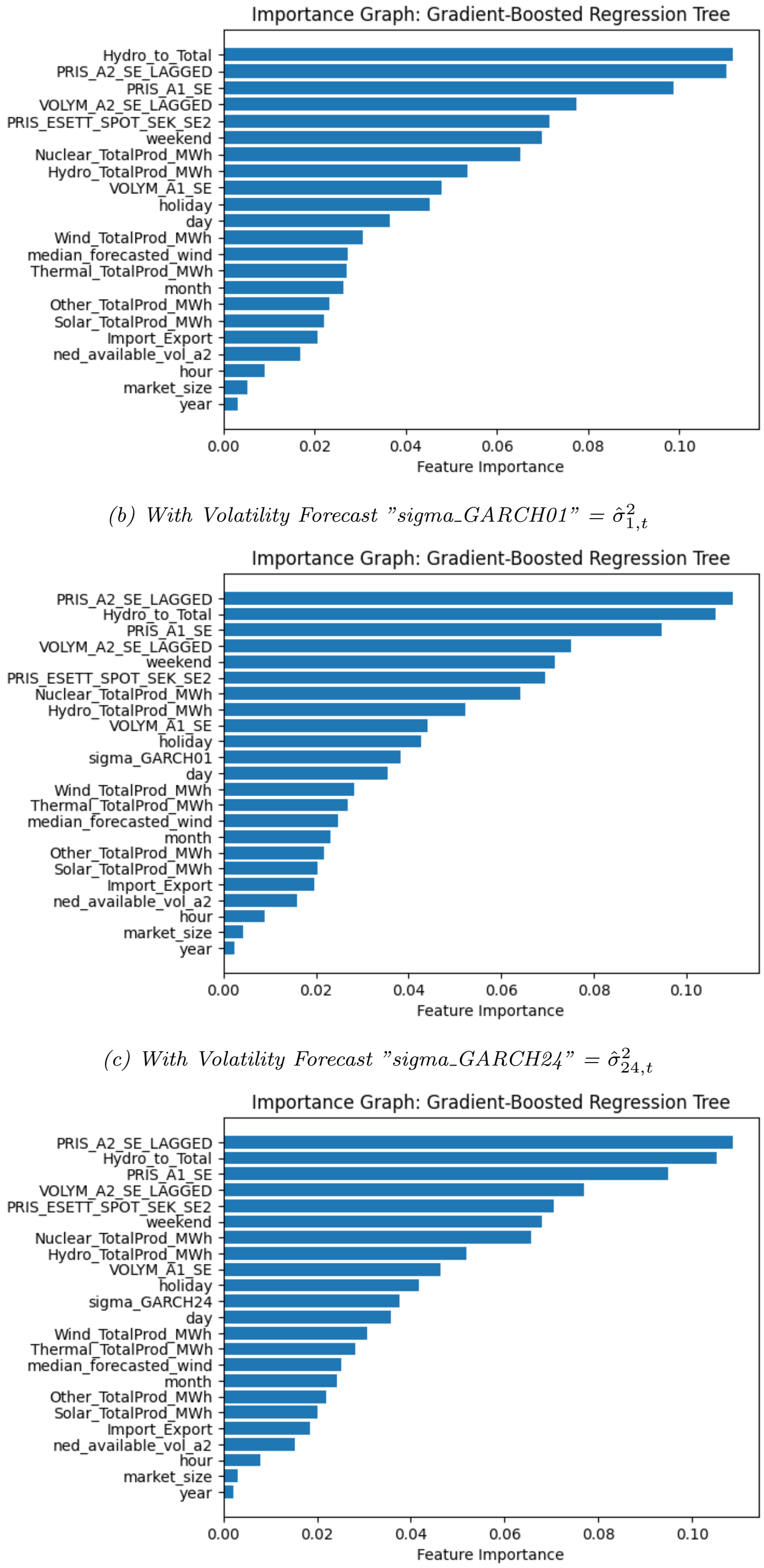
Under the GBRT model, the selection of the top four most important weights did not change. The volatility estimates, named as "sigma_GARCH01" and "sigma_GARCH24" for $\hat{\sigma}^2_{1,t}$ and $\hat{\sigma}^2_{24,t}$, respectively, were not in the top ten of important features meaning that the model did not find the feature to be as meaningful as other features in regards to the fitted data. One of the most important features "PRIS_A2_SE_LAGGED" represents the A2 market price lagged by 24 hours. The other feature, "Hydro_to_Total," represents a ratio of hydroelectricity production to total energy production.

Under the Random Forest model, the volatility estimates were ranked higher than they were in the GBRT models, but were not the most important feature. The distribution of importance values in the Random Forest model appears to be more spread out across the features in comparison to the GBRT, potentially due to the induced randomness in modeling caused by feature subsampling.

Forecast Performance

Table 4.2 represents the performance of the models trained on the three sets of data. Here it is seen the RMSE values of each model for the months between May 2022 and February 2023. I note here that model performances worsened during the months of October 2022 and January 2023, partly due to months containing a high number price spikes. Focusing on the month of October 2022, we can see that the ANN model performed the best in comparison to the other models. We can infer that since the RMSE metric is sensitive to outliers, then the ANN was able to adapt to price spikes better than the other models. I confirm that across all models, the differencs in RMSE values is small except for months with high volatility. Also, models with the volatility forecast $\hat{\sigma}^2_{24,t}$ performed better than their counterparts with lower values of RMSE. Looking at the month January 2023, the models exhibit varying levels of performances. The best model overall appears to be the ANN, with significant improvements in predictions between the data without the volatility estimate and the data with $\hat{\sigma}^2_{24,t}$.

The average of the monthly RMSE values is shown at the bottom of Table 4.2. It appears that the models with lower monthly RMSE values on average are the models using the data set without the volatility. That being said, there are minor differences in averages between the data set containing $\hat{\sigma}^2_{24,t}$ and the data set without a forecast. At the bottom of the table also lies a total mean of all predictions. The largest range of means in the models is between the data sets of the ANN model The result from the data set without the volatility forecast is a mean RMSE of 83.85, while the data set with the volatitlity forecast $\hat{\sigma}^2_{1,t}$ has a mean RMSE 91.73.

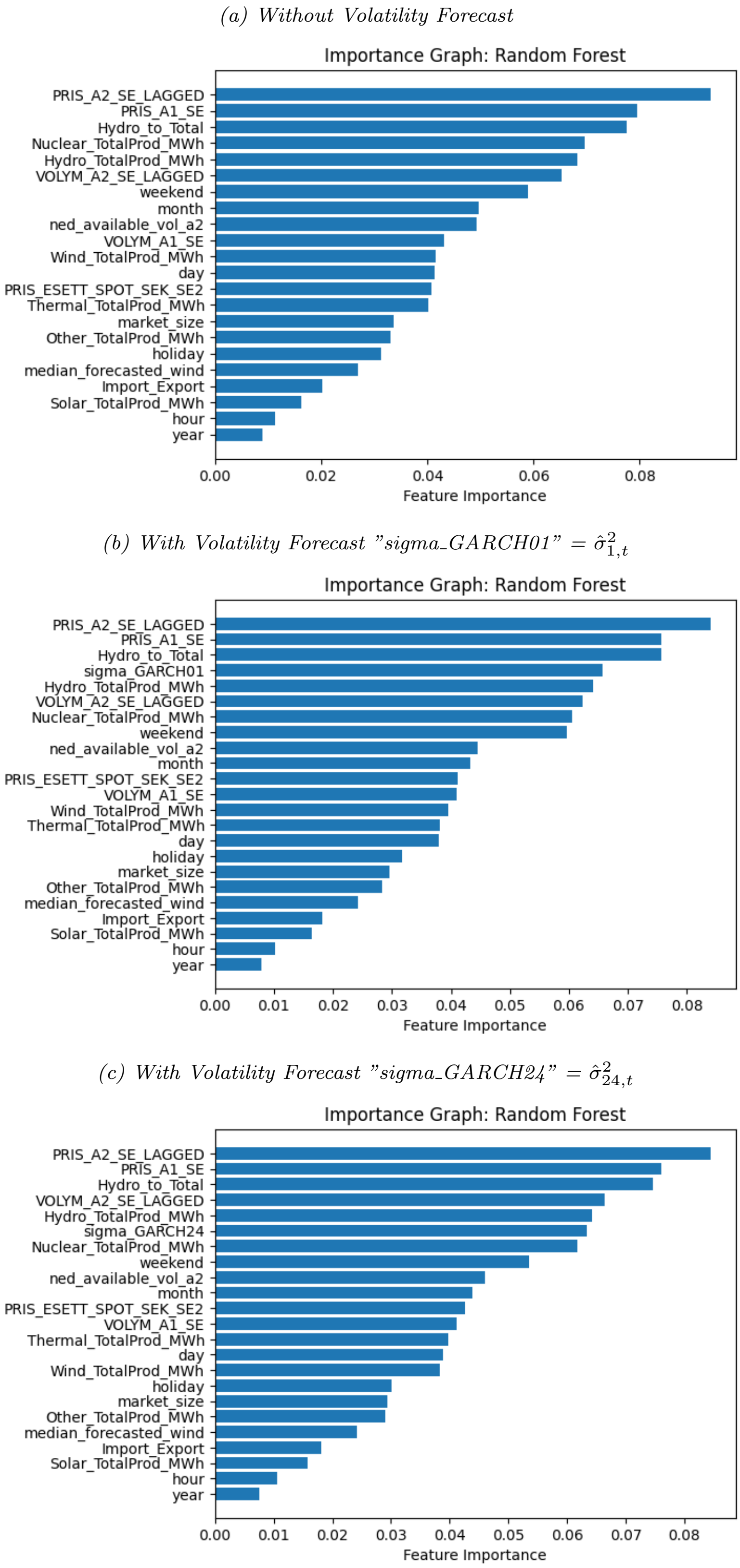


(a) Without Volatility Forecast

(b) With Volatility Forecast "sigma_GARCH01" = $\hat{\sigma}^2_{1,t}$

(c) With Volatility Forecast "sigma_GARCH24" = $\hat{\sigma}^2_{24,t}$

Figure 4.3: Importance Values generated from the GBRT model with and without the volatility forecasts. Values were generated for each time the model was trained daily and then aggregated by taking the mean of all values. The volatility estimates, named as "sigma_GARCH01" and "sigma_GARCH24" for $\hat{\sigma}^2_{1,t}$ and $\hat{\sigma}^2_{24,t}$, respectively, are ranked 11th against the other features in their respective models. The most important feature values are similar across all models.



(a) Without Volatility Forecast

(b) With Volatility Forecast ”sigma_GARCH01” = $\hat{\sigma}^2_{1,t}$

(c) With Volatility Forecast ”sigma_GARCH24” = $\hat{\sigma}^2_{24,t}$

Figure 4.4: Importance Values generated from the Random Forest model with and without the volatility forecasts. The volatility forecasts

RMSE	GBRT			Random Forest			ANN		
Month	No $\hat{\sigma}^2_{h,t}$	With $\hat{\sigma}^2_{1,t}$	With $\hat{\sigma}^2_{24,t}$	No $\hat{\sigma}^2_{h,t}$	With $\hat{\sigma}^2_{1,t}$	With $\hat{\sigma}^2_{24,t}$	No $\hat{\sigma}^2_{h,t}$	With $\hat{\sigma}^2_{1,t}$	With $\hat{\sigma}^2_{24,t}$
May-2022	73.59	81.40	73.43	67.97	69.34	69.98	73.29	73.61	73.48
Jun-2022	42.14	41.55	41.34	40.37	41.62	40.27	44.42	44.63	44.22
Jul-2022	9.07	9.60	9.65	10.20	10.13	10.01	11.46	9.38	11.74
Aug-2022	23.56	23.71	23.63	23.56	24.18	24.36	24.68	24.95	24.42
Sep-2022	42.85	42.56	42.65	41.25	40.18	40.72	46.33	44.52	43.69
Oct-2022	183.66	184.27	181.57	180.88	180.70	178.29	171.88	188.25	166.79
Nov-2022	22.56	23.82	24.07	30.34	28.14	28.24	40.87	33.23	35.07
Dec-2022	70.29	66.60	66.75	65.97	66.14	67.72	66.03	78.43	90.73
Jan-2023	162.85	164.92	164.9	160.09	171.24	166.71	143.91	164.43	143.69
Feb-2023	41.43	43.81	44.97	40.90	43.61	44.32	48.40	50.07	46.12
Monthly Mean	67.20	68.23	67.30	66.15	67.53	67.06	67.13	71.15	68.00
Total Mean	89.00	90.01	88.80	87.08	89.42	88.20	83.85	91.73	84.60

Table 4.2: Monthly RMSE mean values from models with and without volatility estimates, with total RMSE values for all predictions.

MAE	GBRT			Random Forest			ANN		
Month	No $\hat{\sigma}^2_{h,t}$	With $\hat{\sigma}^2_{1,t}$	With $\hat{\sigma}^2_{24,t}$	No $\hat{\sigma}^2_{h,t}$	With $\hat{\sigma}^2_{1,t}$	With $\hat{\sigma}^2_{24,t}$	No $\hat{\sigma}^2_{h,t}$	With $\hat{\sigma}^2_{1,t}$	With $\hat{\sigma}^2_{24,t}$
May-2022	47.6	51.41	48.35	44.37	45.65	45.55	48.09	49.00	48.93
Jun-2022	23.89	23.38	23.63	27.04	27.76	26.59	30.94	30.8	30.45
Jul-2022	5.61	6.12	6.24	7.15	7.24	6.99	8.95	6.13	8.59
Aug-2022	6.16	6.26	6.51	7.34	8.18	8.35	11.37	11.28	11.36
Sep-2022	26.39	26.49	26.17	25.31	25.17	25.59	28.96	27.07	26.71
Oct-2022	80.54	83.69	80.75	83.93	83.32	80.94	87.86	93.91	86.31
Nov-2022	14.30	14.27	14.63	23.54	21.80	21.81	28.5	21.57	23.84
Dec-2022	25.00	23.47	23.94	28.55	29.00	28.81	29.25	35.09	39.28
Jan-2023	56.93	56.25	56.22	65.27	69.32	67.61	67.12	77.95	69.77
Feb-2023	14.99	16.67	17.49	20.79	22.54	23.06	20.58	22.53	22.08
Monthly Mean	30.14	30.8	30.39	33.33	34.00	33.53	36.16	37.53	36.73
Total Mean	30.37	31.03	30.61	33.53	34.20	33.72	36.38	37.79	36.97

Table 4.3: Monthly absolute mean values from models with and without volatility estimates, with total MAE values for all predictions.

The mean absolute errors (MAE) of each model were also analyzed. The equation used to measure the mean absolute errors is

$$\mathrm{MAE} = \frac{1}{T}\Sigma_{i=1}^{T}|y_i - \hat{y}_i| \tag{4.1}$$

They are shown in Table 4.3. Unlike in Table 4.2, the ANN model does not perform the best when measured with MAE. Since RMSE is sensitive to outliers, then we see that ANN performs better on capturing price spikes. During periods of lower volatility, such as July 2022 to August 2022, the ANN does not perform as well as the GBRT model when measured with a less sensitive metric.

From both Tables 4.2 and 4.3, I find that the models with the data sets containing the volatility forecasts had similar or larger total mean values in comparison to the models without the volatility forecasts. However in the month October 2022, which had high volatility, the models with the volatility forecast $\hat{\sigma}^2_{24,t}$ performed similar or better than their counterparts. Results differ in January 2023, the next month with volatile behaviour, where the ANN model tested with volatility estimate $\hat{\sigma}^2_{24,t}$ had a similar RMSE score to the ANN model tested without volatility estimates. GBRT models performed similar to each other under the MAE metric, and outperformed all other models in this regard.

The dispersion of the predictions to actuals are shown in Figures 4.5, 4.6, and 4.7. In combination with the calculated root of covariances and correlations between predictions and actual market prices in Table 4.4, I find that the ANN model has the weakest relationship between its predictions and the actual values, while the GBRT has the strongest relationship. This makes since since GBRT and Random Forest Models return averaged values of market prices based on trained feature values. A GBRT and Random Forest model will not return a value outside the range of the actual values because of this averaging in the leaves of the trees in the ensembles.

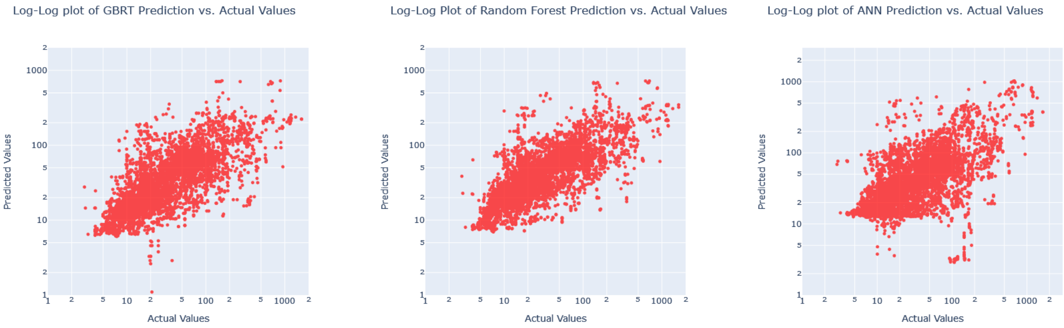


Figure 4.5: Actuals vs. Predictions for all models without $\hat{\sigma}^2_{h,t}$

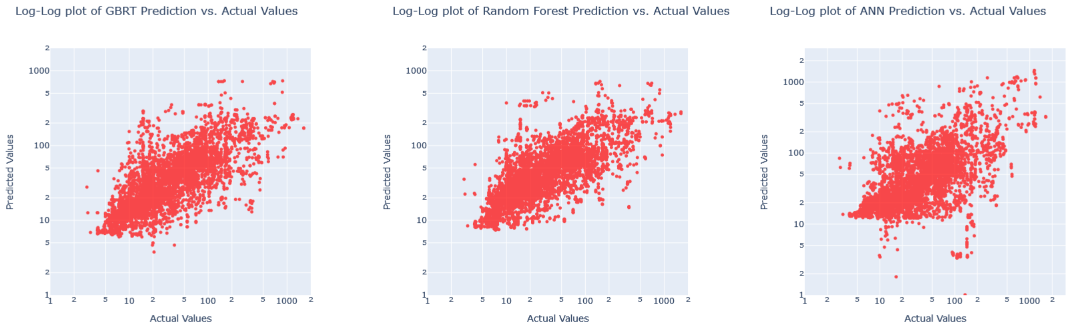


Figure 4.6: Actuals vs. Predictions for all models with $\hat{\sigma}^2_{1,t}$

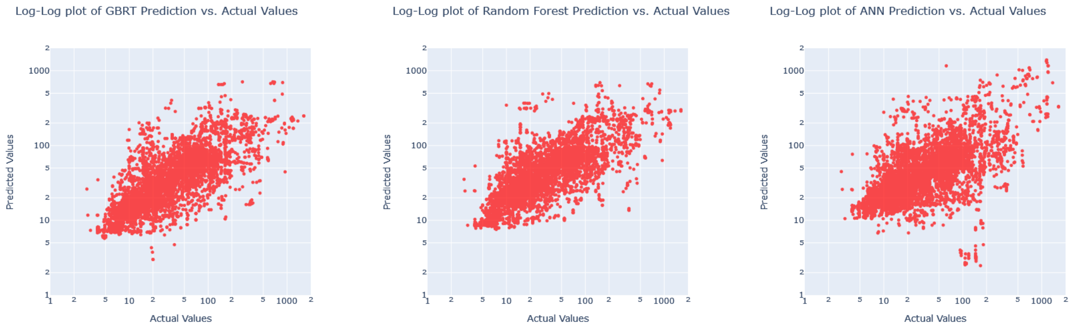


Figure 4.7: Actuals vs. Predictions for all models with $\hat{\sigma}^2_{24,t}$

Model	Data Set	$\sqrt{\text{Covariance}}$	Correlation
GBRT	No $\hat{\sigma}^2_{h,t}$	2409.28	0.81
	With $\hat{\sigma}^2_{1,t}$	2404.36	0.81
	With $\hat{\sigma}^2_{24,t}$	2400.71	0.81
Random Forest	No $\hat{\sigma}^2_{h,t}$	2301.43	0.80
	With $\hat{\sigma}^2_{1,t}$	2365.06	0.80
	With $\hat{\sigma}^2_{24,t}$	2345.02	0.80
ANN	No $\hat{\sigma}^2_{h,t}$	1980.81	0.74
	With $\hat{\sigma}^2_{1,t}$	1996.34	0.68
	With $\hat{\sigma}^2_{24,t}$	1922.76	0.69

Table 4.4: Root covariance values are shown here between the predictions by each respective model and to the actual market prices.

5

Conclusion

The intent of the paper was to produce a hybrid model design that would predict market prices on the FCR-D Down market, using the Swedish ancillary service market as a case study. The paper outlined a price analysis, model theory and methodology used to predict the market prices. The analysis included data sources that would be available in accurate measure to predict the FCR-D Down market prices, followed by a comparison of model performances between the outlined hybrid approach and a machine learning approach. The hybrid approach utilized analysis from a time series forecast combined with a non-parametric machine learning approach. The outlined methodology then described the process of model training via the rolling-origin-recalibration process. Lastly, the methodology introduced the analysis done in tuning each model to then compute the predictions.

Section 4 contains the performance analysis of all models developed for the prediction of FCR-D Down market prices. During the case study, a window size of 30×24 hours was chosen to initialize all models. Thus, predictions are available for the months of May 2022 to February 2023. The resulting performance metrics were analyzed, with the conclusion that there are no observable forecast performance increases with the current hybrid model structure.

5.1 Future Research

There are many ways I could have adjusted the hybrid model design, such as choosing a different variation of the GARCH model. Similarly, a value at risk model can replace the first step in the hybrid model and be used as a feature in the machine learning models. In regards to the chosen machine learning models, the outcome of this paper leads me to conclude that a average of model predictions would be best; such a model would place a large weight on ANN model predictions during periods of high volatility, and place a large weight on the GBRT model during periods of docility. Such a model would then capture best results from both models as described in Section 4. Considering that the time series analysis in this paper revolved around forecasting a volatility estimate for series $r_{h,t+1}$, another approach would be to fit a volatility model on a feature x_{t+1} that is highly correlated with the A2 market price, and use the resulting fitted volatility values as an input in the first step process of the hybrid model design. From a preliminary experiment, forecasting performance was improved by modeling the volatility of A1 market prices.

Bibliography

C. Bergmeir and J. M. Benítez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213, 2012.

T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327, 1986.

T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

R. G. Donaldson and M. Kamstra. An artificial neural network-garch model for international stock return volatility. *Journal of Empirical Finance*, 4(1):17–46, 1997.

E. Ela, M. Milligan, A. Bloom, A. Botterud, A. Townsend, and T. Levin. *Incentivizing Flexibility in System Operations*, pages 95–127. Springer International Publishing, Cham, 2018. ISBN 978-3-319-74263-2. doi: 10.1007/978-3-319-74263-2_5. URL https://doi.org/10.1007/978-3-319-74263-2_5.

W. Enders. *Applied Econometric Time Series*. Wiley Series in Probability and Statistics. Wiley, 2009. ISBN 9780470505397. URL https://books.google.se/books?id=Tc4RPwAACAAJ.

A. Ghalanos. *rugarch: Univariate GARCH models.*, 2022. R package version 1.4-9.

C. Giovanelli, X. Liu, S. Sierla, V. Vyatkin, and R. Ichise. Towards an aggregator that exploits big data to bid on frequency containment reserve market. In *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*, pages 7514–7519. IEEE, 2017.

C. Giovanelli, S. Sierla, R. Ichise, and V. Vyatkin. Exploiting artificial neural networks for the prediction of ancillary energy market prices. *Energies*, 11(7):1906, 2018.

L. R. Glosten, R. Jagannathan, and D. E. Runkle. On the relation between the expected value and the volatility of the nominal excess return on stocks. *The journal of finance*, 48(5):1779–1801, 1993.

E. Kraft, D. Keles, and W. Fichtner. Modeling of frequency containment reserve prices with econometrics and artificial intelligence. *Journal of Forecasting*, 39(8):1179–1197, 2020.

A. Lindholm, N. Wahlström, F. Lindsten, and T. Schön. *Machine Learning: A First Course for Engineers and Scientists*. Cambridge University Press, 2022. ISBN 9781108843607. URL https://books.google.se/books?id=cBhdEAAAQBAJ.

S. Liu, Y. Jiang, Z. Lin, F. Wen, Y. Ding, and L. Yang. Data-driven two-step day-ahead electricity price forecasting considering price spikes. *Journal of Modern Power Systems and Clean Energy*, 2022.

M. Mureddu, G. Caldarelli, A. Chessa, A. Scala, and A. Damiano. Green power grids: How energy from renewable sources affects networks and markets. *PloS one*, 10(9):e0135312, 2015.

F. J. Nogales, J. Contreras, A. J. Conejo, and R. Espínola. Forecasting next-day electricity prices by time series models. *IEEE Transactions on power systems*, 17 (2):342–348, 2002.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

G. P. Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.